M. O. M. JUNE 1984

NEWSLETTER

MICRO OF MONMOUTH

VOL.3 ISSUE 6

JUL MEETINGS

MONDAY JUL 2

8:00 PM
MANALAPAN H.S.

ELECTION OF
OFFICERS

VARIOUS DEMOS

MONDAY JUL 16

8:00 PM
MANALAPAN H.S.

UTILITY DEMO

GAME DEMO

BUSINESS DEMO

WHO'S WHO.

PRESIDENT-Frank Fezza,	201-446-4027
VICE PRES-Irwin Kaplan,	201-566-3335
SECRETARY-Marv Midlarsky,	201-780-2721
TREASURER-John Callaghan,	201-244-4404
PROG.CHAIR-Joe Petrillo,	201-338-8014
LIBRARIAN-James Krutzler,	201-738-7813
LIBRARIAN-Joel Moskow,	201-446-9664
EDITOR-Barry Joseph,	201-536-1932

TABLE OF CONTENTS

PAGE 3

EDITORS MESSAGE

MEETING MINUTES

PAGE 4

NIGHT RAIDERS

PAGE 5

DATABASE II FIX

STORY MACHINE

PAGE 6

LIBRARY REPORT

PAGE 7

Y.P.L.A. DISKS 1-3

** SUPPLEMENT **

ATARI BASIC

Reference Manual Update

M.O.M.
Micro of Monmouth
The Newsletter of the Computer Club of Monmouth,Inc.

Barry Joseph
10 Sherwood Road
Manalapan, N.J. 07726

The Computer Club of Monmouth is a non-profit organization of ATARI Computer Enthusiasts who share a common interest in exchanging information on the programming and operation of the ATARI microcomputer system. The club meets at the Manalapan High School at 8 o'clock on the 1st and 3rd Mondays of each month. The meeting dates are changed or cancelled when a holiday or foul weather causes the school to be closed.

Neither the M.O.M. newsletter or The Computer Club of Monmouth,Inc. is affiliated with ATARI,Inc., Warner Communications or any of their subsidiary companies. All use of the word "ATARI", their model numbers or the Fuji logo, on or between the pages of the M.O.M. newsletter are to be referenced as trademarks of ATARI,Inc.

The M.O.M. newsletter solicits from its members reviews, news, articles, orginial programs, classified ads, and any other material of interest for publication. Any material submitted for publication should be given to the editor at the meetings or have it mailed in.

The material in the M.O.M. newsletter may be freely copied. We request that credit be given to its authors and to M.O.M. This newsletter is sent to paid members of The Computer Club of Monmouth,Inc. and to other users groups on an exchange basis.

Classified ads are free to the members of The Computer Club of Monmouth, Inc.

ATARI COMPUTER
ENTHUSIASTS

EDITORS NOTE

This will be my last issue of the M. O. M. NEWSLETTER and I would like to take some time to reflect over the past years.

About five years ago six people got together and a computer club was formed. The club was very intimate at that time with, less than a dozen members and meetings held in members homes. Everybody got involved and we all learned and grew. Now we have grown much too large for the small intimate meetings but I am happy to say we are still learning and growing. The present leadership has done a very good job to aid us in this growth. The meetings have direction, keeping us informed of industry happenings, teaching us, and exposing us to lots of exciting software. The newsletter has also grown. M.O.M.'s newsletter is now an informative monthly that allows member who are unable to attend meetings to learn some of what they missed. We have also tried to reprint articles of interest from other user groups along with periodic listings of our library programs.

I would like to thank you all for allowing me to be the M.O.M. editor for these many years. I am sure my sucessor will improve on what I had started but he or she will need your support.

I hope to see you at future meetings.

Yours Truely,

Barry S. Joseph

Minutes by Marvin Midlarsky

For 05/07/84

There are now 35 disks available from our club library.

Frank Fezza continued his fine tutorial on DISKTOOL. This second part was an explanation of the development of Hexidecimal numbers.

David Strelitz showed us AUTOTYPE. This is a program that allows the user to set up at least 11 programmable function or PF keys. David is the author of the program and has donated it to our club library, where it will be available shortly to all members.

Steve Marano demoed BRUCE LEE, which is an arcade/adventure game from Datasoft. It is a highly recommended game.

Joe Petrillo demoed SEVEN CITIES OF GOLD, which is an adventure game from Ozark Softscape. It is very highly recommended.

For 05/21/84

The entire meeting was devoted to a fine demonstration by Steve Marano on VisiCalc. VisiCalc, the first of the electronic spreadsheets for home and business computers, is a very useful tool.

NIGHT RAIDERS

Datamost Inc.
8943 Fullbright Ave.
Chatsworth, California 91311

Suggested retail $29.95
Memory required to run 48K

Reviewed by: Michael G. Carovillano

As you settle in at the controls of your air craft you get the erie feeling that this mission is not just one of you normal search and destroy missions. Your orders are to fly across enemy territory and destroy the enemy base. Your secret flight plan will take you across fortified battlefields over fuel depots and train bridges, you will fly in enemy air space which is patrolled by hostile air craft. You will encounter heavy ground fire, which you must avoid, as well as enemy air craft which you must vaporize. You will receive points for destroying enemy defense installations along with fuel depots, which also replenishes your fuel supply. Points are also obtained by attacking and desrtoying tanks, bridges, radar towers, trains, missles, missle silos, space planes, and saucers. To advance to the next level of play you must destroy the enemy base. There are six levels of play, with each level of play making it progressively harder to reach the infamous enemy base. This game utilizes excellent graphic graphic displays coupled with smooth control over your air craft. The game is easy to understand and will give any battle ready pilot many hours of unsurpassed enjoyment. I highly recommend this game to anyone who enjoys the excitement of space flight with the added task of avoiding enemy missles while at the same time inflicting the most amount of damage possible on enemy installations.

"Bad news, Mr. Hunter. Punching out a computer doesn't qualify you for workman's comp."

PRODUCT UPDATE
ATARI HOME COMPUTER SYSTEM

ATARI® BASIC Reference Manual Update

This product update contains a number of corrections and additions to the *ATARI BASIC Reference Manual.*

Page 1. This definition is missing from the **TERMINOLOGY** list:

Floating Point Number: A number containing an integer part, a decimal point, and a fractional part. The total number of significant digits in a floating point number, excluding the exponent, may be either nine or ten. This depends on whether the exponent is an even or odd multiple of 10.

Page 6. This information pertains to the **ARITHMETIC OPERATORS** subtraction and exponentiation:

Note: Avoid negating zero, as this will produce an invalid number. For example, if you type

```
PRINT -0
```

the result will be

```
-0E-<8
```

Note: Since the algorithm used to generate exponents (^) is only an approximation, you cannot obtain integer results with it—for example, 2^2=3.99999996. To correct this, use the following technique:

```
X=2^2
PRINT INT (X+.5)
4
```

Page 7. This Note regards the use of the **LOGICAL OPERATORS**:

Note: Avoid using the statement PRINT A=NOT B, as the results are not predictable. Essentially, any PRINT statement with a NOT operator will be unpredictable.

Page 13. This Note is in reference to **SCREEN EDITING**:

Note: Large amounts of editing may lock up the system. It's recommended that programs under development be stored to cassette or diskette periodically (every 30 or 40 edits) with the SAVE or CSAVE command.

Page 20. This Note regards **ON/GOSUB** statements:

Note: If an ON/GOSUB expression evaluates to a number greater than the number of subroutine entries, then a POP statement will be necessary to clear the stack (see POP command, Section 4).

Page 22. Further information on **RESTORE (RES.)**:

The RESTORE statement will not generate an error if the line number referenced does not exist. Instead it will RESTORE to the next larger line number in the program. Care should be taken to update RESTORE statements when renumbering a BASIC program.

Page 25. Some additional information on using the **INPUT (I.)** statement:

When executing an INPUT from the screen, avoid moving the cursor away from and then back to the same line; otherwise, the wrong data may be input. Specifically, the INPUT prompt will be included in the INPUT string.

If a string of 128-255 characters is INPUT, then RAM locations 1536-1664 will be overwritten. This area is normally reserved for storage of programs or data. (See the ***ATARI Tech Reference Notes.***)

To INPUT strings of more than 127 characters, use the GET command and store the values into a string (see Section 5, OPEN/CLOSE and PUT/GET commands).

Note: The maximum number of characters that can be INPUT from the screen is 120. The maximum for other devices is 255.

Note: Make sure that every INPUT statement has a variable after it; otherwise, unpredictable results may occur.

ATARI®

Page 26. This regards the use of the **LOAD (LO.)** command:

Note: If a program is loaded that is too large for the available memory space, it may give unpredictable results without an error message.

This Note should follow the **LPRINT (LP.)** command description:

Note: An LPRINT command with a semicolon at the end will cause the following LPRINT statement to print on the next 40-column tab. A 40-column printer will move to the next line in such a case. To use the semicolon effectively, use the OPEN statement for the printer, then write to the printer with a PRINT statement (see OPEN/CLOSE and PRINT commands, Section 5).

Page 27. This information pertains to the file spec definition:

Note: Be sure to include the closing quotation marks on a filespec parameter, especially when putting multiple statements on one line. For example,

```
OPEN #1, 4, 0, "D:TEST":STOP
```

will work, but

```
OPEN #1, 4, 0, "D:TEST:STOP
```

will not function correctly.

Page 28. This is an addition to the **POINT (P.)** section:

Note: To update a file, you must open it with a 12 in aexp1.

In the last paragraph under **PRINT (PR. or ?)**, the first sentence should read:

A comma tabs every 10 spaces.

The following sentence should conclude the final paragraph on **PRINT (PR. or ?)**:

However, if the last character to be printed (as in a string with quotation marks) is a [redacted]R or [redacted]U, then the next PRINT will begin at the end of the current line.

This note should then conclude this section on **PRINT (PR. or ?)**:

Note: In rare circumstances data printed to a diskette may have part of the BASIC program embedded in it. If this occurs, retry the operation.

This Note regards the **PUT (PU.)/GET (GE.)** section:

Note: In certain circumstances the GET function may modify other variables within the program. To avoid this, PRINT any number to the screen between each GET.

Page 30. Here is a corrected version of the table—note in particular the correction on cmdno 32:

cmdno	OPERATION	EXAMPLE
3	OPEN	Same as BASIC OPEN
12	CLOSE	Same as BASIC CLOSE
13	STATUS REQUEST	Same as BASIC STATUS
17	DRAW LINE	Same as BASIC DRAWTO
18	FILL	See Section 9
32	RENAME	XIO 32,#1,0,0,"D:TEMP,CAROL"
33	DELETE	XIO 33,#1,0,0,"D:TEMP.BAS"
35	LOCK FILE	XIO 35,#1,0,0,"D:TEMP.BAS"
36	UNLOCK FILE	XIO 36,#1,0,0,"D:TEMP.BAS"
37	POINT	X10 37,#1,A,B
38	NOTE	X10 38,#1,X,Y
254	FORMAT	XIO 254,#1,0,0,"D2:"

ATARI®

Page 33. The last sentence in the paragraph about the **CLOG** function should read:

CLOG(0) through CLOG(1) are inaccurate and should not be used.

Page 34. The last sentence in the paragraph about the **LOG** function should read:

LOG(0) through LOG(1) are inaccurate and should not be used.

Page 38. The last line in the first paragraph should read:

was stored there previously.

Page 39. The first sentence should read:

Upon execution, the screen displays THE SQUARE ROOT OF 10000 IS 100.

In the second paragraph, the last line should read:

number 1000000000.

This is additional information on the **VAL** function:

Only the numeric field will be translated, while the text will be ignored. For example:

VAL("5SUM") = 5

This information pertains to **String Concatenation:**

Note: BASIC cannot move strings of 256-character multiples correctly. String lengths should be checked; if any string contains a multiple of 256 characters, add or subtract one character from the amount to be moved.

In *Figure 7-6*, the correct result of the program on the left is:

BCD#

Page 42. Some additional information on using the **DIM (DI.)** statement:

Make sure that the DIM statement does not contain a space between the string or array name and the left parenthesis of the dimensioned amount; otherwise, the following will happen—

DIM L (10) becomes DIM L10)

—and this variable can no longer be referenced.

Note: The command COM is identical to DIM and may be used in its place.

Page 43. This is an additional Note for the **DIM (DI.)** section:

Note: Due to a discrepancy in boundary checking, arrays of up to 32766 by 32766 in size can be dimensioned. The programmer should size the array ahead of time to ensure that there is no "virtual" storage space.

Additional information on using the **CLR** command:

The second sentence in the last paragraph, beginning "It also clears ...," should be deleted.

The CLR command will not initialize the values in strings and arrays.

ATARI®

Page 45. Here is a corrected version of **TABLE 9.1**:

TABLE 9.1—TABLE OF MODES AND SCREEN FORMATS

Gr. Mode	Mode Type	SCREEN FORMAT Horiz. (Columns)	Vert. (Rows) Split Screen	Vert. (Rows) Full Screen	Number Of Color Registers	Split Screen	RAM Required (Bytes) Full Screen
0	TEXT	40	—	24	1½	—	992
1	TEXT	20	20	24	5	674	672
2	TEXT	20	10	12	5	424	420
3	GRAPHICS	40	20	24	4	434	432
4	GRAPHICS	80	40	48	2	694	696
5	GRAPHICS	80	40	48	4	1174	1176
6	GRAPHICS	160	80	96	2	2174	2184
7	GRAPHICS	160	80	96	4	4190	4200
8	GRAPHICS	320	160	192	1½	8112	8138

Page 49. The last sentence under **PLOT (PL.)** should read:

"The range of points begins at 0 and extends...."

Page 50.

In **TABLE 9.3**, the color PURPLE should be inserted after PINK in the first column, and the number 5 should be inserted after 4 in the second column.

Page 51. The sentence directly under **TABLE 9.4** should read:

"DEFAULT" occurs if no SETCOLOR statement is used.

Page 53. Here is a corrected version of **TABLE 9.5**:

MODE, SETCOLOR, COLOR TABLE

Default Colors	Mode or Condition	SETCOLOR (aexp1) Color Register No.	Color (aexp)	DESCRIPTION AND COMMENTS
		0	Color data	—
LIGHT BLUE	Mode 0 and all text windows	1	actually determines	Character luminance (same color as background)
DARK BLUE		2	character to	Background
		3	be printed.	—
BLACK		4		Border
ORANGE		0	Color data	Character
LIGHT GREEN	Modes 1	1	actually determines	Character
DARK BLUE	and	2	character to be	Character
RED	2	3	printed.	Character
BLACK	(text modes)	4		Background, border
ORANGE		0	1	Graphics point
LIGHT GREEN	Modes 3, 5,	1	2	Graphics point
DARK BLUE	and 7	2	3	Graphics point
	(four-color	—	—	—
BLACK	modes)	4	0	Graphics point (background default), border
ORANGE	Modes 4	0	1	Graphics point
	and 6	—	—	—
	(two-color	—	—	—
	modes)	—	—	—
BLACK		4	0	Graphics point (background default), border
		—	—	—
LIGHT BLUE		1	1	Graphics point luminance (same color as background)
DARK BLUE	Mode 8 (1 color, 2 luminances)	2	0	Graphics point (background default)
		—	—	
BLACK		4	—	Border

ATARI®

Page 54. In *Figure 9-4*, line 80 should read:

80 XIO 18, #6, 12, 0, "S:"

Page 55. This information pertains to **TABLE 9.6:**

In Column 1, # 14, a period, not a bar, shows on the screen.

In Column 3, #'s 92-95 should show a superscripted circled 1 next to their characters.

Page 56. Here is a corrected version of **TABLE 9.7:**

TABLE 9.7—CHARACTER/COLOR ASSIGNMENT

		Column 1 Conversion	Column 2 Conversion	Column 3 Conversion	Column 4 Conversion
MODE 0	[2]SETCOLOR 2	#+32	#+32	#-64	NONE
		POKE 756,224		POKE 756,226	
MODE 1	SETCOLOR 0	#+32	#+32	#-32	#-32
OR	SETCOLOR 1	NONE	#+64	#-64	NONE
MODE 2	SETCOLOR 2	#+160	#+160	#+96	#-96
	SETCOLOR 3	#+128	#+192	#+64	#+128

[2] Luminance controlled by SETCOLOR 1, 0, LUM.

Page 58. The last paragraph should read as follows:

Note that the DATA statement in line 90 ends with 256, which is outside of the designated range. The 256 is.....

In **TABLE 10.1:**

The PITCH VALUE of 193 should have a musical note of "E," not "D."

Page 63. The last line in item 9 should read:

precedence will save a few bytes.

Page 67. In *Figure 11-2*, line 0260 under Data should be:

#2

Page E-1.

The right parentheses are missing after the word "CONSTANT" in Atari Functions of Inverse Cosine, Inverse Secant, and Inverse Cosecant.

Page H-7. Line 160 in the program should read:

160 IF K=125 OR K=155 THEN 180

Page H-8. Line 50 in the program should read:

50 PLOT 0,0:DRAWTO 159, DR

Page 117.

Following COM, "(see DIM)" should be deleted and replaced with "A-1."

Page 118.

Under "Input/Output Devices," Line Printer should be followed by "(P:)," not "(L:)."

Page 119.

"NOTE, 26" is missing from the listing.

ATARI®

Page 120.

INSIDE BACK COVER here is the corrected table:

"STATUS, 29" is missing from the sublisting under "Statement" and also from the regular listing.

MODE, SETCOLOR, COLOR TABLE

Default Colors	Mode or Condition	SETCOLOR (aexp1) Color Register No.	Color (aexp)	DESCRIPTION AND COMMENTS
		0	Color data	—
LIGHT BLUE	Mode 0 and all text windows	1	actually determines character to be printed.	Character luminance (same color as background)
DARK BLUE		2		Background
		3		—
BLACK		4		Border
ORANGE		0	Color data	Character
LIGHT GREEN	Modes 1	1	actually determines	Character
DARK BLUE	and	2	character to be	Character
RED	2	3	printed.	Character
BLACK	(text modes)	4		Background, border
ORANGE		0	1	Graphics point
LIGHT GREEN	Modes 3, 5,	1	2	Graphics point
DARK BLUE	and 7	2	3	Graphics point
	(four-color	—	—	—
BLACK	modes)	4	0	Graphics point (background default), border
ORANGE	Modes 4	0	1	Graphics point
	and 6	—	—	—
	(two-color	—	—	—
	modes)	—	—	—
BLACK		4	0	Graphics point (background default), border
		—	—	—
LIGHT BLUE		1	1	Graphics point luminance (same color as background)
DARK BLUE	Mode 8	2	0	Graphics point (background default)
	(1 color, 2 luminances)	—	—	
BLACK		4	—	Border

©1982 ATARI, INC.

DATABASE.II FIX

BY

Robert H. Brakman, P.E.

One of the most useful pieces of software we have is DATABASE.II on club disk #12. There is, however, an annoying bug in it. Every so often, an extra space gets placed in the string called I$. Since I$ is the basic working file, the extra space throws off sorts, selections, and all kinds of otherwise simple operations.

I think the difficulty arises because ATARI BASIC has problems concatenating strings which are multiples of 256 characters long. No-mind, I found a fix which checks to see if the extra unwanted space was added, and then adjusts. THE CODE FOLLOWS:

```
4026 LG=LEN(I$)
4027 IF J<>0 THEN 4030
4028 IF LG<=NI*IL*(NH+1)THEN 4030
4029 LG=LG-1
4030 I$(LG+1)=A$:NEXT I
```

That's it. ZNow get out there and clean up those lists of people, disks, records, or anything else.

REVIEW ON

STORY MACHINE BY SPINNAKER

DIANE SABBA

Story Machine allows children to create stories, and then see those same stories animated. The stories must be made up from the supplied list of words including nouns, verbs, prepositions and other parts of speech. Children should learn to write correctly, composing sentences, paragraphs and simple stories. The suggested ages for using this package is listed as 5 to 9 years old.

Story Machine is a very good idea for children to see their own created stories come to life. However, it proved to be disappointing. I tested the package out with a couple of six year olds. Within twenty minutes, they were bored.

At first it was exciting to see the sentences come to life. But the supplied list of words is very, very limited. Also, the words that were supplied had to be used in a certain context in order for them to be animated. This completely limited the children's creativety, and their learning to write became more of a chore than the fun it was supposed to be. The excitement quickly died down when the words became boring, the children kept wanting to do it their own creative way, which caused frustration. So back into the box went Story Machine!

LIBRARY REPORT

by
JIM KRUTZLER

I, like many in the M.O.M. club, use the hunt and peck method to type in programs. Of course I consider myself pretty fast after several years of practice. But it is not good enough if you compare this method with the proper way of typing. Well, we do have a typing program that will help each of us to learn the proper way of typing. The program is called "TYPING" and it is found on disks #16 and #19. If you spend about 15 to 20 minutes a night with this program, you'll be surprised how quickly it is to learn typing.

The menu will help you choose any of the nine easy lessons to familiarize your fingers with the keyboard. I like this program because it doesn't bark at you when a mistake is made. Any mistake, including a space, will be high lighted by the inverse character.

Below are some of the screens that you will see when "TYPING" is called up. Notice the inverse character in lesson #1 when a mistake is made.

```
   What lesson do you want ? (1-9) ?1
    LESSON # 1
New letters this lesson :
 Home Keys - A S D F  J K L ; and H
  Place fingers on home keys.
 Press CAPS/LOWR Key.
```

```
ss ll fs jl fs jl aa ;; fa j; fa j; fj
ss ll fs jl fs jl aa ;; fa j; fa j; fj

ss ll fs jl fs jl aa ;; fa j; fa j; fj
ss ll fs jk fs jk
```

```
   What lesson do you want ? (1-9) ?4
    LESSON # 4
New letters this lesson :
  I R . (Period)
 Press CAPS/LOWR Key.
```

```
   What lesson do you want ? (1-9) ?9
    LESSON # 9
New letters this lesson :
  X V 2
 Press CAPS/LOWR Key.
```

REVIEW

YOUNG PEUPLES LOGO ASSO. DISKS 1-3

BY Allan Schwenck

Three disks were submitted by YPLA Program Exchange and reviewed by myself and my two teenagers. We found them very enjoyable and ideal for pre-schoolers and elementry school children. The only requirement in using these disks is the need for a PILOT LANGUAGE CARTRIDGE, a disk drive, 48K for some of the programs and a few simple commands as follows:

```
LOAD D:PROGRAM.EXT  no (") QUOTES used
 RUN
 NEW                      Before loading another program
```

Below from YPLA's catalog is a description of each disk.

ATARI PILOT:

MENU: Load this program for a directory of the programs on the disk

DANGEROUS MISSIONS: Here are the pilot programs from Fred D'ignazio's series of Turtle News articles
ISSION1
BLASTWORM
FACE

PRESCHOOL: A letter recognition game from Ruth Ellsworth.

This is a series of entries to our Graphics Contest 2
CHRIS:
MEG:
JANET:
MICHAEL:
KIRK:

PICTURE: Jenny Chisholm of New Zealand sent us this beautiful drawing program. Make your own pictures.

Ruth Ellsworth of Eugene, Oregon, has sent us a variety of fine programs in addition to her articles.

RANDOM: Random music, color, and lines combine for an interesting audio visual display.

MUSIC & MUSIC2: Here are two short programs to show you how to put mood music to your programs.

PILOTPOK: A timed addition exercise.

DESIGNS: Use your joysticks to create interesting pictures using different shapes.

FAMILYSQUARES: Answer the questions to earn your move in this Tic-Tac-Toe game.

ATARI PILOT #2:

Here are more of Ruth Ellsworth's enjoyable educational programs. Most are written for elementary school children.

RANDOM SPELLING: Enter your own spelling list for drill and practice.

COUNT & MULTIPLY: Good practice in counting and multiplying.

NEXT NUMBER: What comes after the number shown on the screen?

NUMBER BEFORE: What comes before the number appearing on the screen?

CHNEXTNO & CHNOBEF: Similar to the two programs described above but this allows you to select the number of problems you wish to solve.

PLUSONE: Addition practice.

MINUSONE: Subtraction practice.

Arithmetic Practice:
ADDITION:
SUBTRACTION:
MULTIPLICATION:
DIVISION:

TIMED MATH: Select addition, subtraction, multiplication, or division for a timed math drill.

SPELLACE: Spelling practice.

ATARI PILOT #3:

From the Eugene, Oregon, Atari Computer Enthusiasts, comes this Pilot Christmas Disk, 1983.

INSTRUCTION: Instructions for GEOMETRY.

GEOMETRY: An excellent tutorial for teaching geometry concepts. Includes the ability to rotate the figure.

SKETCH.HLP: The instructions for the SKETCH program.

SKETCH: Use your joysticks to create interesting drawings on the screen. Pictures can be saved and recalled.

PIANO.PIL: This program displays the staff and piano keys of notes available in PILOT.

SNBIGLET: A letter matching game for preschoolers.

SNLETTER: A lower case version of the game above. This can also be used to match upper and lower case letters, and for typing practice.

PRETURT: A preschool program designed for the very young. Children can draw using the arrow keys plus others for diagonal lines.

DUNGEON: A tutorial in the simple use of PILOT to create adventure type games.

IN MY OPION, THE PROGRAM "PRESCHOOL" WHICH IS A LETTER RECOGNITION EXERCISE ALLOWES THE CHILD TO GET FAMILAR WITH THE KEYBOARD SEARCHING FOR THE LETTER DISPLAYED ON THE SCREEN AND GETTING A REWARD WHEN CORRECT. FOR THE ELEMENTARY SCHOOL CHILD THERE'S THE MATH, SPELLING AND AN EXCELLANT CREATIVE DRAWING PROGRAM WHICH WE FOUND WAS EXCITING AND FUN. I FOUND THE CREATIVE DRAWING PROGRAMS THE MOST ENJOYABLE , "PICTURE","DESIGN",AND "SKETCH" ALLOWS YOU TO EITHER USE THE DESIGNATED SHAPES PROVIDED OR OBJECTS SUCH AS, TREES,HOUSE,FLOWERS,BOY,GIRL AND OTHER OBJECTS TO CREATE THEIR OWN PICTURE. PROGRAM "SKETCH" ALLOWS THE CHILD TO MAKE THEIR OWN DESIGN USING THE JOYSTICK AND KEYBOARD FOR SHAPING AND COLORING.

OVERALL, WE FOUND THE YPLA PILOT PROGRAMS DISKS WERE EXCELLANT FOR SIMULATED THE IMAGINATION AND INCREASING THE CHILDS LEARNING ABILITIES.

Barry Joseph
10 Sherwood Road
Manalapan, N.J. 07726

FIRST CLASS